Synthetic Data Generation for Cybersecurity Applications: An Exploratory Study of GAN Models

Ricky B. Steinel
Department of Computer Science
William Paterson University
Wayne, NJ 07470, USA
steinelr1@wpunj.edu

Kiho Lim
Department of Computer Science
William Paterson University
Wayne, NJ 07470, USA
limk2@wpunj.edu

Cyril S. Ku
Department of Computer Science
William Paterson University
Wayne, NJ 07470, USA
kuc@wpunj.edu

ABSTRACT

[1]Cybersecurity systems rely heavily on high-quality datasets to train effective intrusion detection mechanisms and malware classification models. However, real-world cybersecurity datasets often face challenges such as limited availability, high annotation costs, privacy risks, and imbalanced class distributions. Generative Adversarial Networks (GANs) have emerged as a promising approach for generating synthetic data that mimics real datasets' statistical properties and structural patterns while preserving privacy. This paper explores multiple GAN variants: CGAN, CycleGAN, BiGAN, DGAN, LSGAN, Tabular GANs, WGAN, WGAN-GP, SeqGAN, and TextGAN, in the context of cybersecurity. Two case studies are conducted: the use of SeqGAN to generate synthetic network traffic based on the NSL-KDD dataset, and the application of TextGAN to produce realistic malicious PHP code snippets. Each model is evaluated for data fidelity, structural validity, and utility in downstream machine learning tasks. The results demonstrate the potential of GANs to enhance data-driven cybersecurity research and operations by simulating attack scenarios and augmenting limited datasets. Future directions include exploring hybrid models, integrating real-time synthesis, and aligning synthetic data generation with evolving threat intelligence paradigms.

KEYWORDS

cybersecurity, GAN, generative adversarial networks, intrusion detection, SeqGAN, synthetic data, TextGAN

1. INTRODUCTION

1.1 Synthetic Data in Cybersecurity

In today's rapidly evolving digital landscape, cybersecurity threats have become increasingly complex, necessitating robust defensive systems grounded in machine learning (ML) and deep learning methodologies. These models depend on large volumes of high-quality labeled data to function effectively. However, obtaining such data is inherently challenging. Real-world cybersecurity datasets often contain sensitive personal or proprietary information, raising ethical and legal concerns [1, 2]. Moreover, the manual annotation of cybersecurity data, such as labeling network traffic or malware samples, requires expert knowledge, making it a time-consuming and costly endeavor. Compounding this issue is the class imbalance prevalent in cybersecurity datasets, where rare attack types are underrepresented, thus impairing model generalizability.

Synthetic data has emerged as a promising solution to address the data scarcity problem. By simulating realistic user behavior and attack vectors, synthetic datasets allow researchers to explore diverse scenarios, including rare and emerging threats. Synthetic data facilitates reproducibility, cost-effective training, and ethical experimentation by decoupling research from private or classified information [3, 4, 5]. It can also address class imbalance, improve adversarial training, and enhance testing under various threat models. The use of synthetic data is particularly compelling in areas like anomaly detection, where rare but critical patterns must be captured effectively [2].

1.2 Generative Adversarial Networks (GANs)

Introduced by Goodfellow et al. [6], GANs consist of a generator and a discriminator network trained in a minimax game framework. The generator attempts to produce realistic data, while the discriminator distinguishes between real and synthetic samples. Over successive iterations, both models improve, leading to high-quality synthetic data generation. GANs have revolutionized image and text synthesis and are now gaining traction in cybersecurity due to their ability to model complex data distributions and generate semantically rich samples [7]. However, training GANs remains challenging due to issues like mode collapse and convergence instability.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICCSS 2025, JUNE 25 - 28, 2025 | Pioneer International University, Nairobi, Keyna

© 2025 Copyright held by the owner/author(s).

1.3 Research Objectives

This paper aims to explore the effectiveness of GAN-based models for generating synthetic data in cybersecurity. Our study evaluates various GAN variants across two use cases: (1) network intrusion detection using the NSL-KDD dataset and SeqGAN, and (2) PHP malware code generation using TextGAN. We compare the models on criteria such as distribution fidelity, structural validity, and applicability to downstream tasks. The goal is to provide a systematic understanding of how different GAN architectures perform in cybersecurity contexts.

2. BACKGROUND

2.1 Related Work

Prior research on synthetic data in cybersecurity has focused on enhancing the effectiveness of machine learning models for threat detection and mitigation. Early efforts leveraged statistical sampling and rule-based simulation techniques to generate synthetic network traffic and log data for testing intrusion detection systems (IDS) under controlled conditions [8]. With the advent of deep generative models, particularly GANs, researchers began producing more realistic and diverse attack simulations. For instance, Lin et al. [9] introduced IDSGAN to generate synthetic intrusion samples that mimic real attack traffic, improving the training of IDS in adversarial environments. Similarly, Douzas & Bacao [10] demonstrated the use of CGANs to generate data for the minority class, significantly improving classification performance in imbalanced scenarios. In the domain of phishing, such as URLs and emails, Karman et al. [11] proposed a GAN network for real-time phishing URL detection. Other papers proposed to use GANs for phishing URLs, websites, and emails, including [12, 13]. Additionally, synthetic data has been used to simulate adversarial behavior, helping identify vulnerabilities in cybersecurity models and systems. These studies collectively highlight the growing consensus that synthetic data is a valuable resource for improving detection accuracy, system robustness, and privacy in cybersecurity research.

While synthetic data generation offers privacy-preserving benefits, it also raises ethical questions [14]. Another concern is the misuse of synthetic data to train more effective cyberattack models, which highlights the dual-use nature of GANs. Ethical deployment requires clear guidelines, rigorous validation, and alignment with institutional data governance policies.

2.2 GAN Overview

GAN framework is a class of generative models that operates by establishing an adversarial relationship between two neural networks: a generator and a discriminator [6]. The generator takes random noise vectors, typically sampled from a uniform or Gaussian distribution, and transforms them into data samples intended to mimic the real data distribution. Meanwhile, the discriminator acts as a binary classifier that distinguishes between real samples (drawn from the true dataset) and fake samples (produced by the generator). The interaction between these two networks is structured as a minimax game, where the generator tries to maximize the probability of the discriminator classifying fake samples as real, while the discriminator seeks to minimize this probability. This dynamic encourages the generator to produce outputs increasingly similar to real data over successive training iterations.

Despite the elegance and theoretical promise of this framework, GANs are notoriously difficult to train. One major issue is training instability, often manifested as non-convergence, where neither network improves meaningfully over time. Another critical issue is vanishing gradients, particularly when the discriminator becomes too strong early in training, leaving the generator with little to no feedback for improvement. Perhaps the most cited problem is mode collapse, wherein the generator converges to producing a limited set of outputs, ignoring large portions of the data distribution and thus compromising diversity.

To address these challenges, a multitude of GAN variants have been proposed [15]. For example, Wasserstein GAN (WGAN) replaces the original loss function with a Wasserstein distance metric to provide smoother gradients and improve training stability [16]. Similarly, WGAN-GP incorporates a gradient penalty term to enforce Lipschitz continuity, further stabilizing the learning process. Other improvements include architectural innovations such as the use of convolutional layers in DCGANs and regularization techniques like feature matching, unrolling GANs, and minibatch discrimination. An overview of many GANs is presented in Section 3.

Recent studies also investigate theoretical aspects of GAN training dynamics, exploring convergence guarantees and equilibrium conditions in the minimax game. These insights have led to the development of game-theoretic GAN training approaches and multi-generator architectures to combat mode collapse and improve output diversity. Despite the progress, successful GAN deployment still requires significant expertise in model design, hyperparameter tuning, and evaluation methodology, particularly in specialized domains like cybersecurity, where data structure and context vary widely.

In summary, GANs represent a powerful but complex tool for generative modeling. Their application in cybersecurity, when properly tailored and stabilized, holds considerable promise for simulating rare attack patterns, preserving privacy, and improving classifier robustness.

2.3 Applications of GANs in Cybersecurity

GANs have found increasing application in cybersecurity for data augmentation, malware traffic generation, anomaly or intrusion detection datasets, adversarial attack generation, and simulation [17]. In IDS, GANs generate synthetic network traffic that mimics malware patterns, which enhances the training and evaluation of ML classifiers [9]. In malware classification, synthetic samples help in building more robust models against obfuscated or evolving threats.

One sizable challenge in cybersecurity is that datasets are often very imbalanced. For example, a dataset may consist of 99.99% normal behavior, with only 0.01% attacks. This will negatively impact the machine learning models, as they will be stunted in detecting rare, malicious events. With GANs, synthetic samples of those rare attacks can be generated, which will help balance the dataset and improve model outlier detection [18].

GANs are also used to simulate adversarial examples—inputs crafted to deceive ML models—thereby exposing vulnerabilities and guiding improvements in model robustness [19]. Another promising area is tabular data generation, where GANs help simulate logs, access records, and system events, preserving feature relationships across complex structured data.

3. GAN VARIANTS FOR CYBERSECURITY APPLICATIONS

3.1 Conditional GAN (CGAN)

CGANs introduce auxiliary information, such as class labels or protocol types, into both the generator and discriminator, enabling conditional data generation. This is particularly valuable in cybersecurity, where control over the type of attack simulated (e.g., DoS, Probe, U2R) can enhance training of classification models [20]. For example, generating network traffic specific to a known attack vector improves model robustness for intrusion detection. CGANs facilitate domain adaptation by generating samples specific to different platforms or services. They can create targeted samples that keep characteristics of each class while exercising variation in those categories.

3.2 CycleGAN and Bidirectional GAN (BiGAN)

CycleGANs are well-suited for domain translation tasks where paired training data is unavailable. For example, translating between normal and malicious traffic profiles can aid in unsupervised anomaly detection [21]. BiGANs include an encoder to infer latent variables for real samples, enabling unsupervised representation learning. This facilitates deeper analysis of the underlying structure of both benign and malicious behaviors [22]. These two types of GANs are great for domain adaptation and transformation. They thrive in unsupervised settings where there is a lack of precise labels, but a need to understand the correlation between two types of data.

3.3 Deep Convolutional GAN (DCGAN)

DCGANs utilize convolutional layers instead of fully connected layers in the GAN architecture, making them highly effective for image-like data generation. Originally applied in computer vision tasks, DCGANs exhibit architectural constraints such as no pooling layers and the use of batch normalization, which improve convergence [23]. In cybersecurity, DCGANs can be adapted to represent visual patterns in network flow or malware binaries converted into grayscale images. For instance, malware binary visualization as images has been used to train convolutional GANs to generate synthetic malware samples that help augment malware detection datasets [24].

3.4 Least Squares GAN (LSGAN)

LSGANs modify the loss function to use least squares error instead of cross-entropy, addressing vanishing gradient problems and producing samples closer to the decision boundary [25]. This yields higher fidelity outputs and better convergence. In cybersecurity, this is helpful for tasks where subtle syntactic structure, such as code generation or packet sequence, must be preserved.

3.5 Tabular GANs (e.g., CTGAN, TableGAN)

Tabular GANs are designed for handling structured data. They excel in generating datasets that preserve intricate relationships between features. CTGAN and TableGAN are specialized GANs designed to generate realistic tabular data. CTGAN employs a mode-specific normalization technique to handle mixed data types and imbalanced modes, which are prevalent in intrusion detection datasets [26]. TableGAN incorporates convolutional architectures for handling structured inputs. These models are particularly valuable in simulating logs, user access patterns, and system event data, thereby improving the realism and diversity of cybersecurity datasets.

3.6 Wasserstein GAN and WGAN-GP

WGANs improve the stability of GAN training by replacing the Jensen-Shannon divergence with the Wasserstein distance, a metric that provides smoother gradients even when distributions do not overlap [16]. WGAN-GP [27] enhances this by enforcing Lipschitz continuity through a gradient penalty, which further stabilizes training, especially in high-dimensional spaces. These variants are useful for generating network flow data or log files with continuous features and high feature correlations.

3.7 Sequential GAN (SeqGAN)

SeqGAN [28] is a specialized class of GANs designed to handle sequential, discrete data such as text, code, or network packet flows. Unlike traditional GANs, which generate continuous data and rely on backpropagation through continuous variables, SeqGAN incorporates a reinforcement learning framework to manage the discrete nature of sequence generation. The generator in SeqGAN is modeled using a Recurrent Neural Network (RNN), such as a Long Short-Term Memory (LSTM), which outputs tokens one at a time. Because direct backpropagation is not feasible through discrete outputs, SeqGAN employs a Monte Carlo search to estimate the expected reward of partially generated sequences, with the discriminator providing feedback as a reward signal. In cybersecurity, SeqGAN has shown promise in generating synthetic network traffic patterns that mimic realistic intrusion attempts, enhancing training datasets for IDS. By simulating rare or emerging attack sequences, SeqGAN enables the creation of more balanced datasets and supports the evaluation of IDS under diverse threat scenarios. It is also useful for generating sequential malware behaviors, command sequences, or even log file events, thereby improving the robustness and generalization of cybersecurity models. We chose this GAN model as one of our experimental prototypes, as described in Section 4.

3.8 TextGAN

TextGAN is a generative adversarial model designed to generate realistic natural language sequences by combining adversarial training with feature matching. Unlike SeqGAN, which emphasizes reward-based sequence generation, TextGAN focuses on learning latent representations of text through a generator and discriminator framework that incorporates Gated Recurrent Units (GRUs) or LSTM layers. The key innovation in TextGAN lies in its use of a feature-matching loss that encourages the generator to produce outputs whose latent features closely align with those of real text, as evaluated by the discriminator. This approach enhances training stability and leads to syntactically coherent and semantically plausible text generation [29]. In the cybersecurity domain, TextGAN has been applied to synthesize realistic code snippets, phishing emails, and malware payloads in languages such as PHP or JavaScript. These synthetic samples are particularly valuable for training and evaluating detection systems, as they enable controlled exploration of attack vectors without the ethical and legal complications of using real malicious content. Additionally, TextGAN can simulate adversarial textual inputs used in social engineering or spam, supporting the development of more resilient natural language-based security tools. Besides SeqGAN, we used TextGAN for our experimental prototype.

4. EXPERIMENTS

As mentioned above, SeqGAN and TextGAN were selected as the primary models for this pilot study due to their suitability for generating sequential and textual synthetic data, respectively. Given the exploratory nature of this initial research, the following section provides a high-level summary of the experimental setup and key findings, without delving into extensive technical details or showing the resulting charts and graphs. Additional experiments and comprehensive analyses are planned as part of the ongoing research agenda, some of them outlined in Section 5.

4.1 Datasets and Preprocessing

This study utilizes two primary datasets: NSL-KDD for network intrusion detection and a corpus of malicious PHP scripts collected from public GitHub repositories. The NSL-KDD dataset is a refined version of the KDD CUP 1999 dataset, offering better class balance and removal of redundant records [30]. We selected a subset of relevant features, including protocol type, service, flag, and duration, and applied min-max normalization to scale continuous variables. The dataset was sourced from Kaggle [31]. For the PHP code dataset, files were tokenized at the character level and segmented into sequences suitable for text generation. We used a dataset consisting of 3,406 PHP code samples, including malicious code snippets. The dataset was sourced from GitHub [32].

4.2 Model Implementation

We implemented SeqGAN and TextGAN using TensorFlow and PyTorch, respectively. SeqGAN models the sequential nature of network traffic using Long Short-Term Memory (LSTM) layers, capturing temporal dependencies in connection sequences. A Monte Carlo search with reward prediction guides adversarial training. For TextGAN, GRUs are used, with feature matching to stabilize the adversarial process. Pre-training was conducted using Maximum Likelihood Estimation (MLE), followed by adversarial refinement. Hyperparameters such as learning rate, batch size, and sequence length were fine-tuned via grid search.

We used these models to generate limited samples of synthetic data for this pilot study. The SeqGAN model generated approximately 10,000 synthetic samples at 16 bytes each. For the PHP TextGAN, the model generated 100 synthetic samples, averaging 1,000 characters at 1 byte per character.

4.3 Evaluation Metrics

Evaluation was conducted using three categories of metrics: (1) Statistical fidelity—assessed via histogram comparisons and Kolmogorov-Smirnov (KS) tests to measure distribution similarity; (2) Syntactic validity—measured by successful parsing rates and token-level perplexity; (3) Downstream utility—evaluated using accuracy and F1 scores of classifiers trained on real vs. synthetic data. Discriminator accuracy and inception scores were also recorded to monitor training progression. Qualitative evaluation involved expert review of generated PHP samples for coherence and structure.

4.4 Key Findings

SeqGAN effectively learned the distribution of network traffic patterns, especially in replicating rare attack types, resulting in improved performance of anomaly detection classifiers. Generated data maintained inter-feature relationships critical for IDS. TextGAN produced syntactically valid PHP code with structural diversity, demonstrating the ability to model real-world malware logic. Pre-training significantly enhanced convergence speed and output quality. Both models benefited from regularization and attention mechanisms in training. Synthetic datasets improved the generalization of downstream models and provided valuable resources for testing under simulated adversarial conditions.

4.5 Sample Results

Figure 1 shows a scatter plot from SeqGAN that visualizes the relationship between source bytes and destination bytes, comparing the patterns of the real data to those of the synthetic data (all scales normalized from zero to one). There is a visible concentration of data points in the lower left quadrant, meaning that a decent amount of network connections use small data transfers. There are also vertical and horizontal streaks, which indicate that there are normal traffic patterns when connections have either high source-to-destination transfers or high destination-to-source transfers. The synthetic data successfully created the distribution patterns. Figure 2 shows the SeqGAN model created for the distribution patterns between duration and source bytes. Figure 3 shows the advanced output from TextGAN, which generates realistic, syntactically correct code. The stealthy malicious scripts include executing system commands, conditionally accessing and modifying files, and constructing dynamic file paths.

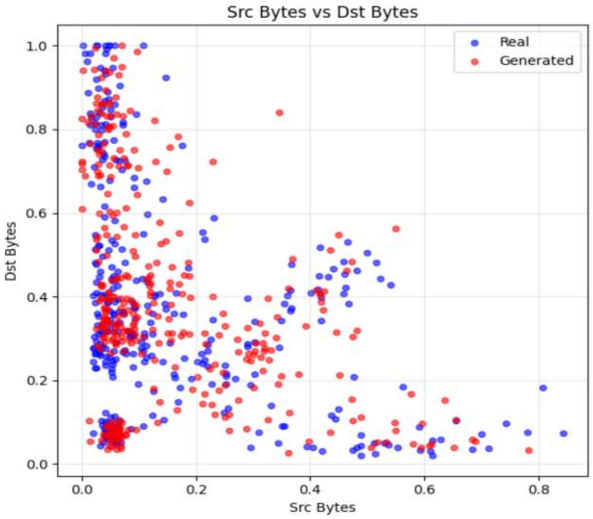


Figure 1: Distribution patterns of real data and synthetic data (source bytes versus destination bytes).

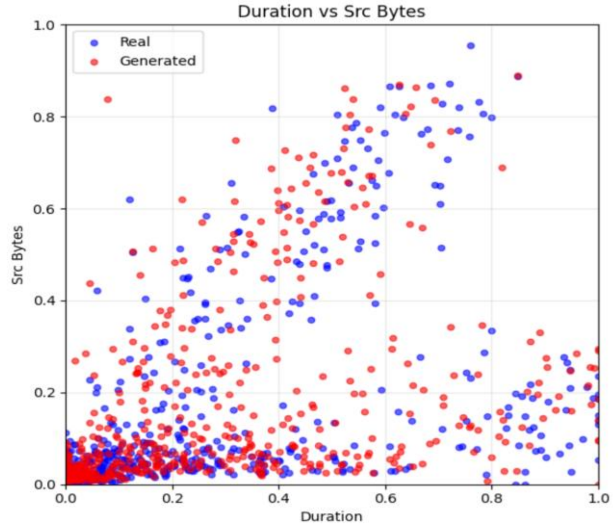
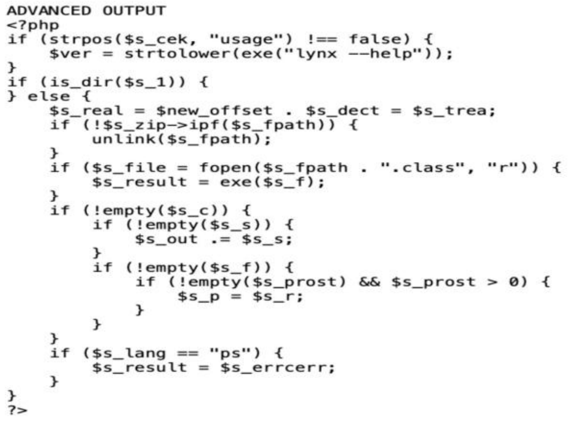


Figure 2: Distribution patterns of real data and synthetic data (duration versus source bytes).



```
ADVANCED OUTPUT
<?php
if (strpos($s_cek, "usage") !== false) {
    $ver = strtolower(exe("lynx --help"));
}
if (is_dir($s_1)) {
} else {
    $s_real = $new_offset . $s_dect = $s_trea;
    if (!$s_zip->ipf($s_fpath)) {
        unlink($s_fpath);
    }
    if ($s_file = fopen($s_fpath . ".class", "r")) {
        $s_result = exe($s_f);
    }
    if (!empty($s_c)) {
        if (!empty($s_s)) {
            $s_out .= $s_s;
        }
        if (!empty($s_f)) {
            if (!empty($s_prost) && $s_prost > 0) {
                $s_p = $s_r;
            }
        }
    }
    if ($s_lang == "ps") {
        $s_result = $s_errcerr;
    }
}
?>
```

Figure 3: Advanced output from TextGAN.

5. CONCLUSION AND FUTURE WORK

This paper presents a systematic exploration of GAN-based models for synthetic data generation in cybersecurity. Our experiments demonstrate that tailored GAN architectures such as SeqGAN and TextGAN are capable of producing high-fidelity synthetic data that preserves statistical and structural properties of real-world datasets. These capabilities enable the augmentation of scarce datasets, the enhancement of anomaly detection systems, and the simulation of sophisticated cyberattacks. While promising, challenges remain in terms of stability, mode diversity, and real-time adaptability. Future research should explore hybrid models that combine GANs with transformers or reinforcement learning, incorporate continual learning frameworks, and develop evaluation metrics that align with operational cybersecurity requirements. Integrating GANs into threat intelligence platforms could further facilitate proactive defense mechanisms in dynamic cybersecurity environments.

As mentioned in Section 4, this research was a pilot study to explore the flexibility and applicability of SeqGAN and TextGAN. For future work, the first thing we plan to do is to increase the size of the synthetic data output. The current experiment only used a typical personal computer to generate synthetic data. We will explore the usage of high-performance computing facilities. Also, we will investigate other GAN models for cybersecurity applications. Further experiments are planned as described below.

5.1 Experiment Extension

Our current experiment was an exploratory study to use limited samples of synthetic data for cybersecurity applications. Future experiments are being planned to include the following.

- Feature-Level Distribution Analysis: Beyond the evaluation of global distribution metrics, we will conduct feature-wise comparisons to ensure that synthetic samples capture the heterogeneity of the input space. For the NSL-KDD dataset, key features such as protocol type, connection duration, and service flags will be analyzed using correlation matrices.
- Adversarial Robustness Testing: To evaluate the robustness of ML models trained on synthetic data, we will introduce adversarial noise to both real and synthetic datasets. Classifiers trained with augmented (real + synthetic) datasets may show increased resilience to adversarial perturbations, particularly in the case of black-box attacks. This supports the hypothesis that GAN-generated samples can enhance model robustness by exposing classifiers to broader data variations [33].

5.2 Limitations and Risks

While promising, GANs for cybersecurity data generation are not without limitations. First, training GANs remains computationally expensive and requires substantial hyperparameter tuning. Second, mode collapse can result in the generator producing low-diversity samples, which may skew downstream analytics [15]. Third, the realism of synthetic data can be hard to quantify, particularly when ground truth labels are not available. Finally, attackers might exploit synthetic datasets to reverse-engineer system assumptions, emphasizing the importance of controlled and transparent deployment.

5.3 Future Trends and Opportunities

Recent advancements in generative modeling suggest that GANs will likely be complemented or even surpassed by architectures such as diffusion models and transformer-based generators in the cybersecurity domain. Moreover, integrating GANs with federated learning enables decentralized, privacy-aware synthetic data generation across edge devices. Real-time synthetic data generation for continuous learning systems and dynamic adversarial simulations represents emerging frontiers. Future research should also prioritize explainability of synthetic instances to align with regulatory frameworks and human-in-the-loop systems.

ACKNOWLEDGMENTS

This work was partially supported by the Student Research Fund of the College of Science & Health at William Paterson University. Additional support was provided by the National Science Foundation under Grant No. 2028011.

REFERENCES

[1] Dunmore, A., Jang-Jaccard, J., Sabrina, F., & Kwak, J. (2023). *A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection*. In *IEEE Access*, vol. 11, pp. 76071-76094, 2023, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10187144

[2] Lu, Y., Chen, L., Zhang, Y., Shen, M., Wang, H., Wang, X., van Rechem, C., Fu, T. & Wei, W. (2021). *Machine Learning for Synthetic Data Generation: A Review,* https://arxiv.org/pdf/2302.04062

[3] El Emam, K., Mosquera, L. & Hoptroff, R. (2020). *Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data,* O'Reilly Media, Inc.

[4] Gursakal, N., Celik, S. & Birisci, E. (2022). *Synthetic Data for Deep Learning: Generate Synthetic Data for Decision Making and Applications with Python and R,* Apress

[5] Patki, N., Wedge, R. & Veeramachaneni, K. (2016). The Synthetic Data Vault, *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA),* https://doi.org/10.1109/DSAA.2016.49

[6] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). *Generative Adversarial Networks,* https://arxiv.org/pdf/1406.2661

[7] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview, *IEEE Signal Processing Magazine,* Volume 35, Issue 1, 53-65. https://ieeexplore.ieee.org/document/8253599

[8] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R. & McClung, D., Weber, D., Wester, S. E., Wyschogrod, D., Cunningham, R. K. & Zissman, M. A. (2000). Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation, *Proceedings DARPA Information Survivability Conference and Exposition (DISCEX'00),* https://doi.org/10.1109/DISCEX.2000.821506

[9] Lin, Z., Shi, Y., Xue, Z. (2022). IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. In: Gama, J., Li, T., Yu, Y., Chen, E., Zheng, Y., Teng, F. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2022. Lecture Notes in Computer Science(), vol 13282. Springer, Cham. https://doi.org/10.1007/978-3-031-05981-0_7

[10] Douzas, G. & Bacao, F. (2018). Effective Data Generation for Imbalanced Learning using Conditional Generative Adversarial Networks, *Expert Systems with Applications,* Volume 91, 2018, Pages 464-471, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2017.09.030

[11] Kamran, S. A., Sengupta, S. & Tavakkoli, A. (2022). *Semi-supervised Conditional GAN for Simultaneous Generation and Detection of Phishing URLs: A Game Theoretic Perspective.* https://arxiv.org/pdf/2108.01852

[12] Gholampour, P. M. & Verma, R. M. (2023). Adversarial Robustness of Phishing Email Detection Models, *IWSPA '23: Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics.* https://dl.acm.org/doi/10.1145/3579987.3586567

[13] Bac, T. N., Duy, P. T. & Pham, V. H. PWDGAN: Generating Adversarial Malicious URL Examples for Deceiving Black-Box Phishing Website Detector using GANs, *2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, Soyapango, El Salvador, 2021, pp. 1-4. doi: 10.1109/ICMLANT53170.2021.9690540

[14] Hao, S., Han, W., Jiang, T., Li, Y., Wu, H., Zhong, C. & Zhou, Z. (2024). *Synthetic Data in AI: Challenges, Applications, and Ethical Implications,* https://arxiv.org/html/2401.01629v1

[15] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, Vicki, Radford, A. & Chen, X. (2016). *Improved Techniques for Training GANs.* https://arxiv.org/pdf/1606.03498

[16] Arjovsky, M., Chintala, S. & Bottou, L. (2017). *Wasserstein GAN.* https://arxiv.org/pdf/1701.07875

[17] Arifin, M. M., Ahmed, M. S., Ghosh, T. K., Udoy, I. A., Zhuang, J. & Yeh, J. H. (2024). *A Survey on the Application of Generative Adversarial Networks in Cybersecurity: Prospective, Direction and Open Research Scopes.* https://arxiv.org/pdf/2407.08839

[18] Zhao, X., Fok, K. W. & Thing, V. L. (2024). *Enhancing Network Intrusion Detection Performance using Generative Adversarial Networks,* https://arxiv.org/pdf/2404.07464

[19] Hu, W. & Tan, Y. (2017). Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN https://arxiv.org/pdf/1702.05983

[20] Mirza, M. & Osindero, S. (2014). *Conditional Generative Adversarial Nets,* https://arxiv.org/pdf/1411.1784

[21] Zhu, J. Y., Park, T., Isola, P. & Efros, A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, *IEEE International Conference on Computer Vision (ICCV),* Venice, Italy, pp. 2242-2251, doi: 10.1109/ICCV.2017.244.

[22] Donahue, J., Krahenbuhl, P. & Darrell, T. (2016). *Adversarial Feature Learning,* https://arxiv.org/pdf/1605.09782

[23] Radford, A., Metz, L. & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/pdf/1511.06434

[24] Saxe, J. & Berlin, K. (2015). *Deep Neural Network Based Malware Detection Using Two Dimensional Binary Program Features,* https://arxiv.org/pdf/1508.03096

[25] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z. & Smolley, P. S. "Least Squares Generative Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2813-2821, doi: 10.1109/ICCV.2017.304

[26] Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019). *Modeling Tabular Data Using Conditional GAN,* https://arxiv.org/pdf/1907.00503

[27] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. (2017). *Improved Training of Wasserstein GANs.* https://arxiv.org/pdf/2302.04062

[28] Yu, L., Zhang, W., Wang, J. & Yu, Y. (2017). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence,* Vol. 31, No. 1. https://doi.org/10.1609/aaai.v31i1.10804

[29] Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D. & Carin, L. (2017). *Adversarial Feature Matching for Text Generation,* https://arxiv.org/pdf/1706.03850

[30] Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set, *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications (CISDA'09),* Ottawa, Canada. ISBN: 978-1-4244-3763-4

[31] https://www.kaggle.com/datasets/hassan06/nslkdd

[32] https://github.com/nb1b3k/php-malware-samples

[33] Zenati, H., Romain, M., Foo, C. S., Lecouat, B. & Chandrasekhar, V. R. (2018). *Adversarially Learned Anomaly Detection,* https://arxiv.org/pdf/1812.02288